

REMARKS

Claims 1-39 are pending in the Application and all were rejected in the Office action mailed July 10, 2007. No claims are amended in this response. Claims 1, 17 and 32 are independent claims. Claims 2-16, 18-31 and 33-39 depend, respectively, from independent claims 1, 17 and 32. The Applicants respectfully request reconsideration of pending claims 1-39, in light of the following remarks.

Rejections of Claims

Rejections Under 35 U.S.C. §103(a)

Claims 1-4, 6, 7, 10-19, 21, 22, and 25-39 were rejected under 35 U.S.C. §103(a) as being unpatentable over US 2004/0031029 by Lee et al. (hereinafter "Lee"), in view of Meyerson (US Patent 6,976,251). The Applicants respectfully traverse the rejection.

The Applicants respectfully submit that the Examiner has failed to establish a case of prima facie obviousness for at least the reasons provided below. M.P.E.P. §2142 clearly states that "[t]he examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness." The M.P.E.P. §2142 goes on to state that "[t]o establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure."

With regard to claims 1, 17 and 32, Applicants respectfully submit that the proposed combination of references, fails to teach, suggest or disclose, at least, "...a plurality of update agents resident in the electronic device, wherein the update agent employed is selected to correspond to a type of update information received by the electronic device from the at least one of the plurality of servers...", as recited in Applicants' claim 1;

“...selecting at least one of a plurality of update agents resident in the electronic device... wherein each of the plurality of update agents is arranged to process a corresponding type of update information received...”, as recited in Applicants’ claim 17; and “...code comprising a plurality of update agents selectable to cause processing of a corresponding type of received update information ... wherein an update agent is selected to perform an update based upon the type of the received update information...”, as recited in Applicants’ claim 32.

Applicants respectfully note that although the instant Office action withdrew the rejection of claims 1, 17 and 32 under the combination of Lee and Lee Chen-Yin et al. (US Patent 6,915,325, hereinafter “Lee CY”) as set forth in the Office action of October 24, 2006, the Office continues to rely on the Lee reference. The instant Office action fails to address Applicants’ arguments regarding the alleged teachings of Lee. (See Applicants’ response filed April 24, 2007 at pages 11-14) Instead, the instant Office action simply repeats the same arguments regarding the Lee reference that were presented in the Office action of October 24, 2006, including the conclusory statement that “...Lee teaches all aspects of claim 1, but he does not mention ‘update agent is selected to correspond to a type of update information.’” (Office action at page 3) While the Applicants appreciate recognition in the Office action that “...[Lee] does not mention ‘update agent is selected to correspond to a type of update information’...”, Applicants respectfully submit that this admitted shortcoming of Lee is not all that Lee does not mention.

Applicants respectfully maintain that Lee fails to teach or suggest, at least, “...a plurality of update agents resident in the electronic device, wherein the update agent employed is selected to correspond to a type of update information received by the electronic device from the at least one of the plurality of servers...”, as recited in Applicants’ claim 1; “...selecting at least one of a plurality of update agents resident in the electronic device ... wherein each of the plurality of update agents is arranged to process a corresponding type of update information received...”, as recited in Applicants’ claim 17; and “...code comprising a plurality of update agents selectable to cause processing of a corresponding type of received update information ... wherein an update agent is selected

to perform an update based upon the type of the received update information...", as recited in Applicants' claim 32.

The Office action again cites paragraphs [0009] and [0032] of Lee and, based upon these identified portions, alleges that Lee teaches "...all aspects of claim 1, but he does not mention 'update agent is selected to correspond to a type of update information', specifically...". (Office action at page 3) With regard to claim 17, the Office action states that "...Claim 17 is a method version of claim 1, therefore see claim 1 rejection..." (Office action at pages 8-9) With respect to claim 32, the Office action states that "...Both Lee and Meyerson's disclosures teach the features of claim 32, see claims 1, 2, and 3 rejections." (Office action at page 11) The Applicants respectfully disagree with what Lee and Meyerson allegedly teach.

The Lee reference states, at paragraph [0009],

"The invention relates, in one embodiment, to a computer-implemented method for updating a plurality of software components disposed on a plurality of networked devices, the plurality of networked devices being interconnected if a computer network. The method includes ascertaining from a database first update parameters associated with a first networked device of the plurality of networked devices. The method also includes sending via the network the first update parameters to a first local update agent disposed at the first networked device. The method further includes obtaining, using the first local update agent and the first update parameters, a first update file for updating software in the first networked device. Additionally, the method includes updating, using the first local update agent and the first update file, the software in the first networked device."

Applicants respectfully submit that the above portion of Lee simply describes a method in which a database is used to ascertain update parameters associated with one of a plurality of networked devices. The networked device contains an update agent which uses the update parameters to obtain an update file. The update agent then uses the update file to update software in the networked device. Applicants respectfully submit that

this portion of Lee, however, is silent with respect to an electronic device having a plurality of update agents.

The Office action also relies on Lee at paragraph [0032], which states:

“These and other features and advantages of the invention may be better understood with reference to the drawings and discussions that follow. To facilitate discussion, FIG. 1 shows a simplified prior art network 102, which includes an administrative console 104. Administrative console 104 is coupled via the network to a plurality of networked devices such as servers 106, 108, and 110. In the example of FIG. 1, servers 106, 108, and 110 represent servers running, for example, the Windows, Netware, and Linux operating systems respectively to illustrate that different networked devices may employ different software components therein. Administrative console 104 is also shown Coupled to desktop computers 112 and 114, as well as to a lap top computer 116 and a printer 118. In reality, a computer network may be coupled to any number and type of networked device in any topology (e.g., mesh, ring, star, and the like) to enable appropriate networked devices to communicate with one another.”

Applicants respectfully submit that this portion of Lee simply states that an administrative console is coupled to a plurality of networked devices that are servers running an operating system and that may employ different software components, and that a computer network may be coupled in any topology to enable communication between the networked devices. Applicants respectfully submit, however, that this portion of Lee also fails to teach anything with respect to an electronic device having a plurality of update agents.

Applicants wish to point out the teachings of the second (paragraph [0010]) of the two paragraphs of Lee’s “Summary”, which states:

“The invention relates, in another embodiment, to an arrangement for updating a plurality of software components disposed on a plurality of networked devices, the plurality of networked devices being interconnected in a computer

network. The arrangement includes a database for storing update parameters associated with the plurality of networked devices. The update parameters includes at least a name for a first one of the plurality of software components and a version number associated with the name for the first one of the plurality of software components. The arrangement further includes a plurality of local update agents, each of the plurality of local update agents being disposed at **one** of the plurality of networked devices. There is also included a software update engine configured to send individual sets of the update parameters to individual ones of the plurality of local update agents at the plurality of network devices, wherein a first one of the plurality of local update agents disposed at a first one of the plurality of networked devices is configured to obtain, upon receiving a first one of the individual sets of the update parameters, a first update file using the first one of the individual sets of update parameters. The first update file represents a file containing data for updating a first one of the plurality of software components disposed at the first one of the plurality of networked devices.

(emphasis and underline added)

Applicants respectfully submit that this portion of Lee simply teaches a plurality of software components disposed on a plurality of networked devices in a computer network that includes a database for storing update parameters, and a plurality of local update agents. Lee makes it clear that each of the plurality of local update agents is disposed at one of the plurality of networked devices. Applicants respectfully submit that there is a conspicuous absence of any statement or suggestion by Lee that any of the networked devices has more than one update agent. Although the above portion of Lee teaches a “plurality of networked devices” and a “plurality of software components disposed on a plurality of networked devices”, neither this nor any other portion of Lee teaches or suggests that any networked device has more than one “local update agent”. In addition, the drawings in the Lee reference, including FIG. 6 and related text that teach a “local update agent 336”, fail to teach or suggest anything regarding the existence of more than one update agent in a networked device.

Applicants respectfully maintain that the Office action has failed to specifically identify any teachings of Lee that correspond to the “plurality of update agents resident in the electronic device” recited in Applicants’ claim 1. Applicants respectfully submit that neither the text of paragraphs [0009] and [0032] of Lee, which were specifically identified in the Office action, nor any other portion or figure of Lee teaches or suggests “...a plurality of update agents resident in the electronic device...”, as recited in Applicants’ claims 1 and 17; and “...code resident in and executable by the electronic device, the code comprising a plurality of update agents...”, as recited in claim 32. Applicants respectfully submit that Lee fails to provide any teaching or suggestion to one of ordinary skill in the art regarding the existence or use of more than one update agent resident in an electronic device.

Therefore, for at least the above reasons, Applicants respectfully maintain that the Lee reference fails to teach or suggest, at least, “...a plurality of update agents resident in the electronic device,...”, as recited in Applicants’ claims 1 and 17, and “...code comprising a plurality of update agents ...”, as recited in Applicants’ claim 32.

The Applicants respectfully submit that Meyerson also fails to teach or suggest, at least, “...a plurality of update agents resident in the electronic device,...”, as recited in Applicants’ claims 1 and 17, and “...code comprising a plurality of update agents ...”, as recited in Applicants’ claim 32. However, the Office action asserts, at page 3, that ‘...Meyerson teaches [update agent is selected to correspond to a type of update information] in an analogous prior art. In Meyerson’s column 4, lines 38-50, “After the software update query is sent, the software update information is downloaded in block 14. In the simplest case, where **the update agent corresponds to a single software program** (*a plurality of update agents resident in the electronic device*), the software update information may simply be a “yes” or “no” telling the agent whether a software update is available and whether a criticality check program is available. In the more general case, however, the software update information will include substantial additional information for multiple software programs. In the preferred implementation of the invention, the software update information will include a field telling the update agent

whether a criticality check program is available for each software update.” (emphasis and italicized text added in the Office action)

The Office action fails to set forth any reasoning as to how the Examiner arrived at the conclusion that the cited text “...In the simplest case, where the update agent corresponds to a single software program, the software update information may simply be a "yes" or "no" telling the agent whether a software update is available and whether a criticality check program is available...”, teaches or suggests “...a plurality of update agents resident in the electronic device....”

Applicants respectfully submit that the Office has misinterpreted/misunderstood the teachings of Meyerson. Applicants respectfully point the Examiner to Meyerson at column 3, line 66 to column 4, line 55, which reads (cited portion in bold):

Referring to FIG. 1, the intelligent update agent of this invention begins its program flow at start point 10. The intelligent update agent may be designed in accordance with this invention so that the program is started manually, however, in the preferred implementation, the start point 10 includes program code with a timer that periodically restarts program flow to regularly check for the existence of software updates. The intelligent update agent software may also be provided with a user setting to change between manual and automatic starting and to set the period of time between automatic starts.

Once flow has started, it proceeds to block 12 where the intelligent update agent sends a software update query comprising a request for software update information. The software update query is sent over a computer network, such as the Internet. In one implementation of the invention, the intelligent update agent is designed specifically for and may be incorporated into particular software. The agent will then include the address of a particular location maintained by the software publisher that corresponds to the software. In this type of implementation, the software update query contains very little information. The software publisher knows that all queries arriving at a particular location are requests for software update information corresponding to that location.

In a more general implementation of the invention, the update agent may be designed by a particular software publisher to handle all of that publisher's software. In this design, the request for software update information may carry with it software identification information comprising a list of all of the software by that software publisher that is installed on the user's computer.

In yet another implementation of the invention, the update agent may be designed for handling multiple software publishers. In this case the software update query may contain software publisher information that identifies the software publishers of software installed on the user's computer. Alternatively, it may carry both software publisher information and software identification information or only software identification information.

After the software update query is sent, the software update information is downloaded in block 14. **In the simplest case, where the update agent corresponds to a single software program, the software update information may simply be a "yes" or "no" telling the agent whether a software update is available and whether a criticality check program is available.** In the more general case, however, the software update information will include substantial additional information for multiple software programs. In the preferred implementation of the invention, the software update information will include a field telling the update agent whether a criticality check program is available for each software update. In addition, the software update information will include an address field telling the update agent where to find the criticality check program, if one is available, an address field telling the update agent where to find the software update itself, and a field containing an initial criticality rating for the software update.

The software update information may provide information about only software identified in the update query. However, the preferred design is for the update query to identify the software publishers having software installed on the users computer and for the software update information to include a list of software updates available for software published by the publishers identified in the initial software update query.

Those of skill in the art will recognize that the amount of information contained in the software update query from the user's computer and the amount of information carried in the software update information response may be varied. This allows the designer of the update agent software to distribute the workload between the update agent making the request and the server providing the response and to control the amount of traffic required between the user's computer and the server computer. However, after the software update information is downloaded, the intelligent update agent will have sufficient information to determine whether a software update is available for each software program being managed by the update agent.

(emphasis and underlining added)

Applicants respectfully submit that the above portion of Meyerson, which includes those teachings of Meyerson presumably chosen by the Office as the teachings most relevant to Applicants' claims, simply describes three possible implementations of an update agent (underlined). Meyerson teaches a first implementation in which the intelligent update agent is designed specifically for and may be incorporated into particular software, a second more general implementation of the invention in which the update agent may be designed by a particular software publisher to handle all of that publisher's software, and a third implementation in which the update agent may be designed for handling multiple software publishers. Applicants respectfully submit, however, that neither these portions of Meyerson, nor any other portions or figures of Meyerson teach or suggest the use of a plurality of update agents in an electronic device, and that the Office has failed to show where Meyerson teaches or suggests, at least, "...a plurality of update agents resident in the electronic device,...", as recited in Applicants' claims 1 and 17, and "...code comprising a plurality of update agents ...", as recited in Applicants' claim 32. Therefore, Applicants respectfully submit that Meyerson also does not teach or suggest at least this aspect of Applicants' claims.

Based at least on the above, Applicants respectfully submit that both Lee and Meyerson fail to teach or suggest, at least, "...a plurality of update agents resident in the

electronic device,...", as recited in Applicants' claims 1 and 17, and "...code comprising a plurality of update agents ...", as recited in Applicants' claim 32, and that the proposed combination of Lee and Meyerson, therefore, by definition cannot teach "...a plurality of update agents resident in the electronic device,...", as recited in Applicants' claims 1 and 17, and "...code comprising a plurality of update agents ...", as recited in Applicants' claim 32.

The Applicants now turn to the admitted shortcoming of Lee, and the alleged teachings of Meyerson. As previously discussed, the Office action has recognized that Lee "...does not mention 'update agent is selected to correspond to a type of update information.'" (Office action at page 3) However, to overcome this deficiency of the Lee reference, the Office action asserts:

'...Meyerson teaches [update agent is selected to correspond to a type of update information] in an analogous art. In Meyerson's column 4, lines 38-50, "After the software update query is sent, the software update information is downloaded in block 14. In the simplest case, where **the update agent corresponds to a single software program** (*a plurality of update agents resident in the electronic device*), the software update information may simply be a "yes" or "no" telling the agent whether a software update is available and whether a criticality check program is available. In the more general case, however, the software update information will include substantial additional information for multiple software programs. In the preferred implementation of the invention, the software update information will include a field telling the update agent whether a criticality check program is available for each software update." And Meyerson's column 4, lines 4-16, "it proceeds to block 12 where the intelligent update agent sends a software update query comprising a request for software update information. The software update query is sent over a computer network, such as the Internet. In one implementation of the invention, the intelligent update agent is designed specifically for and may be incorporated into particular software."

(emphasis and italics in original)

To clarify these alleged teachings of Meyerson, the Applicants show the cited portions underlined and in context, starting at column 3, line 66 to column 4, line 55:

Referring to FIG. 1, the intelligent update agent of this invention begins its program flow at start point 10. **The intelligent update agent may be designed in accordance with this invention so that the program is started manually, however, in the preferred implementation, the start point 10 includes program code with a timer that periodically restarts program flow to regularly check for the existence of software updates.** The intelligent update agent software may also be provided with a user setting to change between manual and automatic starting and to set the period of time between automatic starts.

Once flow has started, it proceeds to block 12 where the intelligent update agent sends a software update query comprising a request for software update information. The software update query is sent over a computer network, such as the Internet. In one implementation of the invention, the intelligent update agent is designed specifically for and may be incorporated into particular software. The agent will then include the address of a particular location maintained by the software publisher that corresponds to the software. In this type of implementation, the software update query contains very little information. The software publisher knows that all queries arriving at a particular location are requests for software update information corresponding to that location.

In a more general implementation of the invention, the update agent may be designed by a particular software publisher to handle all of that publisher's software. In this design, the request for software update information may carry with it software identification information comprising a list of all of the software by that software publisher that is installed on the user's computer.

In yet another implementation of the invention, the update agent may be designed for handling multiple software publishers. In this case the software update query may contain software publisher information that identifies the software publishers of software installed on the user's computer. Alternatively, it may carry both software publisher information

and software identification information or only software identification information.

After the software update query is sent, the software update information is downloaded in block 14. In the simplest case, where the update agent corresponds to a single software program, the software update information may simply be a "yes" or "no" telling the agent whether a software update is available and whether a criticality check program is available. In the more general case, however, the software update information will include substantial additional information for multiple software programs. In the preferred implementation of the invention, the software update information will include a field telling the update agent whether a criticality check program is available for each software update. In addition, the software update information will include an address field telling the update agent where to find the criticality check program, if one is available, an address field telling the update agent where to find the software update itself, and a field containing an initial criticality rating for the software update.

The software update information may provide information about only software identified in the update query. However, the preferred design is for the update query to identify the software publishers having software installed on the users computer and for the software update information to include a list of software updates available for software published by the publishers identified in the initial software update query.

Those of skill in the art will recognize that the amount of information contained in the software update query from the user's computer and the amount of information carried in the software update information response may be varied. This allows the designer of the update agent software to distribute the workload between the update agent making the request and the server providing the response and to control the amount of traffic required between the user's computer and the server computer. However, after the software update information is downloaded, the intelligent update agent will have sufficient information to determine whether a software update is available for each software program being managed by the update agent.

(emphasis and underlining added)

Applicants respectfully submit that the portions of Meyerson shown above in bold clearly show that the update agent of Meyerson is either started manually, or periodically by a timer, to check for software updates. Meyerson also states that "...the intelligent update agent sends a software update query comprising a request for software update information." Therefore, Meyerson clearly teaches that the update agent is active and running before a request for a software update is sent, and before a software update is received. Because the update agent of Meyerson is started before a software update is received, the Meyerson reference does not teach an update agent "...wherein the update agent employed is selected to correspond to a type of update information received by the electronic device from the at least one of the plurality of servers...", in accordance with Applicants' claim 1. Applicants respectfully submit that for the same reason, Meyerson does not teach or suggest, at least, "...selecting at least one of a plurality of update agents resident in the electronic device to modify a first version of one of software and firmware in the electronic device to produce an updated version, wherein each of the plurality of update agents is arranged to process a corresponding type of update information received from the at least one of a plurality of servers...", as recited in Applicants' claim 17; and "...code resident in and executable by the electronic device, the code comprising a plurality of update agents selectable to cause processing of a corresponding type of received update information, to update a related code portion of the first version of code to an updated version...", as recited in Applicants' claim 32. Therefore, Applicants respectfully submit that Meyerson fails to remedy the admitted deficiency of Lee, and that the combination of Lee and Meyerson fails to teach or suggest this aspect of Applicants' claims.

Based at least upon the above, Applicants respectfully submit that the proposed combination of Lee and Meyerson fails to teach or suggest all of the limitations of Applicants' claims 1, 17, and 32, as required by M.P.E.P. §2142, that the Office has failed

to establish a *prima facie* case of obviousness, and that a rejection of Claims 1, 17, and 32 under 35 U.S.C. §103(a) cannot be maintained.

Therefore, Applicants believe that claims 1, 17 and 32 are allowable over the proposed combination of Lee and Meyerson, for at least the reasons set forth above. Applicants respectfully submit that claims 1, 17 and 32 are independent claims, and that claims 2-16, 18-31 and 32-39 are dependent claims depending from allowable independent claims 1, 17 and 32, respectively. Because claims 1, 17 and 32 are allowable over the proposed combination of Lee and Meyerson, Applicants respectfully submit that dependent claims 2-16, 18-31 and 32-39 are also allowable, for at least the reasons set forth above. Applicants respectfully request, therefore, that the rejection of claim 1-7, 10-22 and 25-39 under 35 U.S.C. 103(a), be withdrawn.

Claims 8 and 23 were rejected under 35 U.S.C. §103(a) as being unpatentable over Lee, in view of Meyerson, and further in view of Kikinis (US 5,708,775). The Applicants respectfully traverse the rejection. Applicants respectfully submit that claims 8 and 23 depend from independent claims 1 and 17, respectively. Applicants believe that claims 1 and 17 are allowable over the proposed combination of references, in that Kikinis fails to overcome the deficiencies of Lee and Meyerson, set forth above. Because claims 1 and 17 are allowable over the proposed combination of references, Applicants respectfully submit that dependent claims 8 and 23 are also allowable, as well. Therefore, Applicants respectfully request that the rejection of claims 8 and 23 under 35 U.S.C. 103(a), be withdrawn.

Conclusion

In general, the Office Action makes various statements regarding claims 1-39 and the cited references that are now moot in light of the above. Thus, Applicants will not address such statements at the present time. However, the Applicants expressly reserve the right to challenge such statements in the future should the need arise (e.g.,

Appln. No.: 10/807,694
Filed: March 24, 2004
Reply to Office Action of July 10, 2007
Response dated October 5, 2007

if such statements should become relevant by appearing in a rejection of any current or future claim).

The Applicants believe that all of pending claims 1-39 are in condition for allowance. Should the Examiner disagree or have any questions regarding this submission, the Applicants invite the Examiner to telephone the undersigned at (312) 775-8000 to resolve any issues.

A Notice of Allowability is courteously solicited.

Respectfully submitted,

Dated: October 5, 2007

/Kevin E. Borg/
Kevin E. Borg
Reg. No. 51,486

Hewlett-Packard Company
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400